IN THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the application.

Claims 1-8, 10-12 and 14-16 have been amended and claim 17 has been canceled as follows:

Listing of Claims:

Claim 1 (currently amended): A "Device for Reducing the Width of Graph" which reduces the width of the Binary Decision Diagram for Characteristic Function (BDD_for_CF), where BDD_for_CF is a characteristic function χ (X,Y) defined in Equation (1), $X=(x_1,\cdots,x_n)(n\in N$, N is a set of natural numbers) denotes input variables, $Y=(y_0, \cdots, y_{m-1})(m\geqq 2, m\in N)$ denotes the output variables of a multiple-output logic function $F(X)$, and $F(X)=(f_0(X),\cdots,f_{m-1}(X))$ is an incompletely specified function to the output including don't care, comprising:

(A) "Means to Store Node Table" storing the node table which is the table of node data that consists of the labels of variables and pairs of edges $e_0(v_i)$ *and* $e_1(v_i)$, where the labels of variables are labels given to the variables z_i $(z_i\in(X\cup Y))$ corresponding to said each non-terminal node v_i in the BDD_for_CF of the multiple-output logic function $F(X)$, and a pair of edges

e_0 (v_i) and e_1 (v_i) that points the next transition child node(s) when the input values of z_i $(z_i\in(X\cup Y))$ are 0 and 1;

(B) "Means to Find the Dividing Lines" setting the height of the partition *lev*

which partitions BDD_for_CF represented by said node table stored in said "Means to Store Node Table";

(C) "Means to Generate Column Functions" generating a column function which represents the column of the decomposition chart derived by the functional decomposition from said node table stored in said "Means to Store Node Table", where the decomposition is obtained by partitioning said BDD_for_CF by said height of the partition *lev* set by said "Means to Find the Dividing Lines"; and

(D) "Means to Reconstruct Assigned BDD" which assigns the constants to the don't care in the compatible column functions of column function generated by said "Means to Generate Column Functions", and consequently makes these compatible column functions to the identical column functions (hereafter, assigned column functions), and reconstructs said BDD_for_CF using the new assigned column function, and finally updates the node table in said "Means to Store Node Table"[[.]] ,

[Equation 1]

$$\chi(X,Y) = \bigwedge_{i=0}^{m-1} \{\bar{y}_i f_{i\_0} \vee y_i f_{i\_1} \vee f_{i\_d}\} \qquad (1)$$

where $f_{i\_0}, f_{i\_1}, f_{i\_d}$ are the OFF function, the ON function and the DC function defined in Equation (2), respectively[[.]] ,

[Equation 2]

$$f_{i\_0}(X) = \begin{cases} 1 & (X \in f_i^{-1}(0)) \\ 0 & (\text{otherwise}) \end{cases}, f_{i\_1}(X) = \begin{cases} 1 & (X \in f_i^{-1}(1)) \\ 0 & (\text{otherwise}) \end{cases}, f_{i\_d}(X) = \begin{cases} 1 & (X \in f_i^{-1}(d)) \\ 0 & (\text{otherwise}) \end{cases} \qquad (2)$$

Claim 2 (currently amended): The "Device for Reducing the Width of Graph" according to Claim 1, wherein:

the device further comprises;

(E) "Means to Store Compatible Graphs" storing the compatible graph as a table of function node data, that is a table of column function labels of said each function node and the data of compatible edges connected to the function node, where the compatible graph is a graph which has said column functions as nodes (function nodes), and wherein a pair function nodes corresponding to the column functions compatible each other are connected by an edge (compatible edge);

(F) "Means to Generate Compatible Edges" which selects the pair of compatible column function from the set of column functions corresponding to said each function node data, stored in said "Means to Store Compatible Graphs", and then adds a compatible edge which connects these function nodes with function node data corresponding to these compatible column functions, and finally updates the function node data stored in said "Means to Store Compatible Graphs"; and

(G) "Means to Generate Cliques" covering nodes with the minimum number of complete subgraphs (cliques) for all nodes in said compatible graph, and then generating clique data of function node set contained within the clique; and

said "Means to Generate Column Functions" generates column functions corresponding to each edge of nodes at said height of the partition *lev* set by said "Means to Find the Dividing Lines" from said node table stored in "Means to Store Node Table", and then generates said function node data having column function labels corresponding

to these column functions, and then stores in said "Means to Store Compatible Graphs", and

said "Means to Reconstruct Assigned BDD" reconstructs said BDD_for_CF by making some column functions to the identical assigned column functions by assigning the constants to don't care of the column functions corresponding to each function node contained in the clique data produced by said "Means to Generate Cliques", and updates said node table in said "Means to Store Node Table".

Claim 3 (currently amended): The "Device for Reducing the Width of Graph" according to Claim 1 [[or 2]], wherein:

said "Means to Find the Dividing Lines" sets the height of the partition *lev* sequentially from the height of the child node of the root node in BDD_for_CF represented by said node table stored in said "Means to Store Node Table", towards the low height, and

said "Means to Reconstruct Assigned BDD" reconstructs sequentially in said each height of the partition *lev* set by said "Means to Find the Dividing Lines".

Claim 4 (currently amended): A "Device for Logic Synthesis" which generates look-up tables (LUTs) of the data for constructing logic circuits corresponding to said multiple-output logic function $F(X)$ from the BDD_for_CF of the multiple-output logic function $F(X)=(f_0(X), \cdots, f_{m-1}(X))$ with input variables $X=(x_1, \ldots, x_n)$ $(n \in N)$, comprising:

(A) "Means to Store Node Table" storing BDD_for_CF representing the characteristic function $\chi(X, Y)$ (where $Y=(y_0, \cdots, y_{m-1})$ $(m \geqq 2, m \in N)$ denotes output

variables of $F(X)$) defined in Equation (3), as the node table which is the table of node data that consists of the labels of variables and pairs of edges $e_0(v_i)$ and $e_1(v_i)$, where said multiple-output logic function $F(X)=(f_0(X), \cdots, f_{m-1}(X))$ is a completely specified function, the labels of variables are labels given to the variables z_i ($z_i \in (X \cup Y)$) corresponding to said each non-terminal node v_i in the BDD_for_CF, and a pair of edges e_0 (v_i) and e_1 (v_i) that points the next transition child nodes when the input values of z_i ($z_i \in (X \cup Y)$) are 0 and 1;

(B) "Means to Store LUTs" storing said LUTs;

(C) "Means to Find the Dividing Lines" setting the height of the partition lev which partitions BDD_for_CF represented by said node table stored in said "Means to Store Node Table";

(D) "Means to Reduce by Shorting" executing shorten-processing that is the processing to replace the edge $e_c(v_k)$ that points the node v_j among two edges $e_0(v_k)$ and $e_1(v_k)$ of the parent node v_k of the node v_j, by the edge $e_b(v_j)$ other than the edge $e_a(v_j)$ of the node v_j, in the case that the terminal node related to $\chi(X,Y)=0$ pointed by the edge $e_a(v_j)$ of either the edges $e_0(v_j)$ or $e_1(v_j)$ of the node v_j, about the node data of the node v_j related to the variable $y_r(\in Y)$ representing output and the parent node v_k of the node v_j, where the nodes v_j and v_k are contained in the subgraph B_0 including the root node among the node data of non-terminal nodes stored in said "Means to Store Node Table", in the case of partitioning BDD_for_CF to the two subgraphs B_0 and B_1 at the partition line in said height of the partition lev;

(E) "Means to Measure the Width of BDDs" which counts the number of the edges that point the child nodes of the non-terminal nodes, whose height is smaller than

said height of the partition *lev*, among the edges which are the non-terminal nodes in BDD_for_CF to which said shorten-processing by said "Means to Reduce by Shorting" is applied and which belong to the non-terminal nodes whose height is larger than said height of the partition *lev* (where the edges pointing the same node is counted as one, and the edge to point the constant 0 is disregarded), and produces the number as the width W at the partition line in said height of the partition *lev*;

(F) "Means to Compute the Intermediate Variables" calculating the number of intermediate variables u following Equation (4), using the width W produced by said "Mean to Measure the Width of BDDs";

(G) "Means to Generate LUTs" which generates LUTs from the node data and stores said LUTs in said "Means to Store LUTs", for the non-terminal nodes which belong to the subgraph B_0 including the root node, among the non-terminal nodes stored in said "Means to Store Node Table", in the case of partitioning said BDD_for_CF into two subgraphs at the partition line in said height of the partition *lev*; and

(H) "Means to Re-construct BDDs" which generates a binary tree which has the same number of control inputs as the number of intermediate variables u which is calculated by said "Means to Compute the Intermediate Variables", and reconstructs the BDD_for_CF by replacing the node data of non-terminal nodes in subgraph B_0 of BDD_for_CF stored in said "Means to Store Node Table" by the node data representing said binary tree, and updates the node table stored in said "Means to Store Node Table" by the node data of the non-terminal nodes in said reconstructed BDD_for_CF[[.]]

[Equation 3]

$$\chi(X,Y) = \bigwedge_{i=0}^{m-1} (y_i \equiv f_i(X)) \tag{3}$$

[Equation 4]

$$u = \lceil \log_2 W \rceil \tag{4}$$

Claim 5 (currently amended): The "Device for Logic Synthesis" according to Claim 4, wherein said "Means to Store Node Table" stores the BDD_for_CF as a node table, where said BDD_for_CF is a graph that represents the characteristic function $\chi(X,Y)$ (where $Y=(y_0, \cdots, y_{m-1})$ ($m \geqq 2, m \in N$) denotes the output variables of $F(X)$) defined in Equation (5), with said multiple-output logic function $F(X)=(f_0(X), \cdots, f_{m-1}(X))$ of an incompletely specified function that includes don't cares in outputs, said node table is the table of the node data that consists of the labels of variables and pairs of edges $e_0(v_i)$ and $e_1(v_i)$, said labels of variables are labels given to the variables z_i ($z_i \in (X \cup Y)$) corresponding to said each non-terminal node v_i in the BDD_for_CF, and said pair of edges

$e_0(v_i)$ and $e_1(v_i)$ that points the next transition child nodes when the values of z_i ($z_i \in (X \cup Y)$) are 0 and 1[[.]],

[Equation 5]

$$\chi(X,Y) = \bigwedge_{i=0}^{m-1} \{\bar{y}_i f_{i\_0} \vee y_i f_{i\_1} \vee f_{i\_d}\} \tag{5}$$

where $f_{i\_0}, f_{i\_1}, f_{i\_d}$ are the OFF function, the ON function and the DC function defined in Equation (6), respectively[[.]];

[Equation 6]

$$f_{i\_0}(X)=\begin{cases}1 & (X\in f_i^{-1}(0))\\ 0 & (\text{otherwise})\end{cases},\ f_{i\_1}(X)=\begin{cases}1 & (X\in f_i^{-1}(1))\\ 0 & (\text{otherwise})\end{cases},\ f_{i\_d}(X)=\begin{cases}1 & (X\in f_i^{-1}(d))\\ 0 & (\text{otherwise})\end{cases} \quad (6)$$

.

Claim 6 (currently amended): ~~The "Device for Logic Synthesis" according to Claim 5,~~ A "Device for Logic Synthesis" which generates look-up tables (LUTs) of the data for constructing logic circuits corresponding to said multiple-output logic function $F(X)$ from the BDD_for_CF of the multiple-output logic function $F(X)=(f_0(X), \cdots, f_{m-1}(X))$ with input variables $X=(x_1, \ldots, x_n)$ $(n\in N)$, comprising:

(A) "Means to Store Node Table" storing BDD_for_CF representing the characteristic function $\chi(X, Y)$ (where $Y=(y_0, \cdots, y_{m-1})$ $(m\geqq 2,\ m\in N)$ denotes output variables of $F(X)$) defined in Equation (3), as the node table which is the table of node data that consists of the labels of variables and pairs of edges $e_0(v_i)$ and $e_1(v_i)$, where said multiple-output logic function $F(X)=(f_0(X), \cdots, f_{m-1}(X))$ is a completely specified function, the labels of variables are labels given to the variables z_i $(z_i\in(X\cup Y))$ corresponding to said each non-terminal node v_i in the BDD_for_CF, and a pair of edges $e_0(v_i)$ and $e_1(v_i)$ that points the next transition child nodes when the input values of z_i $(z_i\in(X\cup Y))$ are 0 and 1;

(B) "Means to Store LUTs" storing said LUTs;

(C) "Means to Find the Dividing Lines" setting the height of the partition *lev* which partitions BDD_for_CF represented by said node table stored in said "Means to Store Node Table";

(D) "Means to Reduce by Shorting" executing shorten-processing that is the processing to replace the edge $e_c(v_k)$ that points the node v_j among two edges $e_0(v_k)$ and $e_1(v_k)$ of the parent node v_k of the node v_j, by the edge $e_b(v_j)$ other than the edge $e_a(v_j)$ of the node v_j, in the case that the terminal node related to $\chi(X,Y)=0$ pointed by the edge $e_a(v_j)$ of either the edges $e_0(v_j)$ or $e_1(v_j)$ of the node v_j, about the node data of the node v_j related to the variable $y_r(\in Y)$ representing output and the parent node v_k of the node v_j, where the nodes v_j and v_k are contained in the subgraph B_0 including the root node among the node data of non-terminal nodes stored in said "Means to Store Node Table", in the case of partitioning BDD_for_CF to the two subgraphs B_0 and B_1 at the partition line in said height of the partition *lev*;

(E) "Means to Measure the Width of BDDs" which counts the number of the edges that point the child nodes of the non-terminal nodes, whose height is smaller than said height of the partition *lev*, among the edges which are the non-terminal nodes in BDD_for_CF to which said shorten-processing by said "Means to Reduce by Shorting" is applied and which belong to the non-terminal nodes whose height is larger than said height of the partition *lev* (where the edges pointing the same node is counted as one, and the edge to point the constant 0 is disregarded), and produces the number as the width W at the partition line in said height of the partition *lev*;

(F) "Means to Compute the Intermediate Variables" calculating the number of intermediate variables u following Equation (4), using the width W produced by said

"Mean to Measure the Width of BDDs";

(G) "Means to Generate LUTs" which generates LUTs from the node data and stores said LUTs in said "Means to Store LUTs", for the non-terminal nodes which belong to the subgraph B_0 including the root node, among the non-terminal nodes stored in said "Means to Store Node Table", in the case of partitioning said BDD_for_CF into two subgraphs at the partition line in said height of the partition *lev*; and

(H) "Means to Re-construct BDDs" which generates a binary tree which has the same number of control inputs as the number of intermediate variables u which is calculated by said "Means to Compute the Intermediate Variables", and reconstructs the BDD_for_CF by replacing the node data of non-terminal nodes in subgraph B_0 of BDD_for_CF stored in said "Means to Store Node Table" by the node data representing said binary tree, and updates the node table stored in said "Means to Store Node Table" by the node data of the non-terminal nodes in said reconstructed BDD_for_CF

[Equation 3]

$$\chi(X,Y) = \bigwedge_{i=0}^{m-1} (y_i \equiv f_i(X)) \qquad (3)$$

[Equation 4]

$$u = \lceil \log_2 W \rceil \qquad (4)$$

wherein said "Means to Store Node Table" stores the BDD_for_CF as a node table, where said BDD_for_CF is a graph that represents the characteristic function $\chi(X,Y)$ (where $Y=(y_0, \cdots, y_{m-1})$ ($m \geqq 2$, $m \in N$) denotes the output variables of $F(X)$) defined in Equation (5), with said multiple-output logic function $F(X)=(f_0(X), \cdots, f_{m-1}(X))$ of an incompletely specified function that includes don't cares in outputs, said node table is the table of the node data that consists of the labels of variables and pairs of edges $e_0(v_i)$ and $e_1(v_i)$, said labels of variables are labels given to the variables z_i ($z_i \in (X \cup Y)$) corresponding to said each non-terminal node v_i in the BDD_for_CF, and said pair of edges

$e_0(v_i)$ and $e_1(v_i)$ that points the next transition child nodes when the values of z_i ($z_i \in (X \cup Y)$) are 0 and 1,

[Equation 5]

$$\chi(X,Y) = \bigwedge_{i=0}^{m-1} \{\bar{y}_i f_{i\_0} \vee y_i f_{i\_1} \vee f_{i\_d}\} \tag{5}$$

where $f_{i\_0}, f_{i\_1}, f_{i\_d}$ are the OFF function, the ON function and the DC function defined in Equation (6), respectively,

[Equation 6]

$$f_{i\_0}(X) = \begin{cases} 1 & (X \in f_i^{-1}(0)) \\ 0 & (\text{otherwise}) \end{cases}, f_{i\_1}(X) = \begin{cases} 1 & (X \in f_i^{-1}(1)) \\ 0 & (\text{otherwise}) \end{cases}, f_{i\_d}(X) = \begin{cases} 1 & (X \in f_i^{-1}(d)) \\ 0 & (\text{otherwise}) \end{cases} \tag{6}$$

wherein

the device comprises the "Device for Reducing the Width of Graph" according to ~~any one of claims 1 to 3~~ claim 1, and

said "Means to Reduce by Shorting" reduces the width of BDD_for_CF represented by said node table stored in said "Means to Store Node Table" by said "Device for Reducing the Width of Graph", and then performs said shorten-processing on the updated node table.

Claim 7 (currently amended): The "Device for Logic Synthesis" according to ~~any one of claims 4 to 6~~ claim 4, wherein the device comprises:

"Means to Decide the Ordering of Output Variables" deciding the order π of elements of said multiple-output logic function $F(X)$ to minimize the value of T represented in Equation (7), where $\pi=(\pi[0], \cdots, \pi[m-1])$ ($\pi[i]=j$ represents that f_j is the i'th element) is the order of the logic functions $f_0(X), \cdots, f_{m-1}(X)$ that are elements of said multiple-output logic function $F(X)$, and $supp(f_j)$ is the set of input variables that influence the logic function f_j ($\in F(X)$);

"Means to Decide the Ordering of all the Variables" deciding the order of the variables $y_j(\in Y)$ representing the outputs and input variables x_i ($\in X$) in the order P that satisfies Equation (8); and

"Means to Generate BDDs" which generates node data of the BDD_for_CF according to the order P decided in said "Means to Decide the Ordering of all the Variables", and then stores in said "Means to Generate BDDs"[[.]]

[Equation 7]

$$T = \sum_{k=0}^{m-1} \left| \bigcup_{l=0}^{k} \mathrm{supp}(f_{\pi[l]}) \right| \tag{7}$$

[Equation 8]

$$P = \left(\mathrm{supp}(f_{\pi[0]}), y_{\pi[0]}, \mathrm{supp}(f_{\pi[1]}) - \mathrm{supp}(f_{\pi[0]}), y_{\pi[1]}, \mathrm{supp}(f_{\pi[2]}) - \left(\sum_{k=0}^{1} \mathrm{supp}(f_{\pi[k]})\right), y_{\pi[2]}, \right.$$
$$\left. \cdots, \mathrm{supp}(f_{\pi[m-1]}) - \left(\sum_{k=0}^{m-2} \mathrm{supp}(f_{\pi[k]})\right), y_{\pi[m-1]}\right) \tag{8}$$

Claim 8 (currently amended): A Method to Reduce the Width of Graph which reduces the width of the BDD_for_CF, in the system comprising "Means to Store Node Table" which stores the node table which is the table of node data that consists of the labels of variables and pairs of edges $e_0(v_i)$ and $e_1(v_i)$, where BDD_for_CF is a characteristic function $\chi(X,Y)$ defined in Equation (9), $X=(x_1,\cdots,x_n)(n\in N$, N is a set of natural numbers) are input variables, $Y=(y_0, \cdots, y_{m-1})(m\geqq 2, m\in N)$ denotes the output variables of a multiple-output logic function $F(X)$, $F(X)=(f_0(X), \cdots ,f_{m-1}(X))$ is an incompletely specified function to the output including don't care, the labels of variables are labels given to the variables z_i ($z_i\in(X\cup Y)$) corresponding to said non-terminal node v_i in the BDD_for_CF of the multiple-output logic function $F(X)$, and a pair of edges e_0 (v_i) *and* e_1 (v_i) that points the next transition child nodes when the input values of z_i ($z_i\in(X\cup Y)$) are 0 and 1, comprising the steps of:

a) a "Step to Find Dividing Lines" setting the height of the partition *lev* which partitions BDD_for_CF represented by said node table stored in said "Means to Store

Node Table";

b) a "Step to Generate Column Functions" generating a column function which represents the column of the decomposition chart derived by the functional decomposition from said node table stored in said node table in said "Means to Store Node Table", where the decomposition is obtained by partitioning said BDD_for_CF by said height of the partition *lev* set in said "Step to Find Dividing Lines"; and

c) a "Step to Reconstruct Assigned BDD" assigning the constants to the don't care in the compatible column functions of the column function generated in said "Step to Generate Column Functions", and consequently making these compatible column functions to the identical column functions (hereafter, assigned column functions), and reconstructing said BDD_for_CF using new assigned column functions, and finally updating the node table in said "Means to Store Node Table"[[.]],

[Equation 9]

$$\chi(X,Y) = \bigwedge_{i=0}^{m-1} \{\bar{y}_i f_{i\_0} \vee y_i f_{i\_1} \vee f_{i\_d}\} \tag{9}$$

where $f_{i\_0}, f_{i\_1}, f_{i\_d}$ are the OFF function, the ON function and the DC function defined in Equation (10), respectively[[.]],

[Equation 10]

$$f_{i\_0}(X) = \begin{cases} 1 & (X \in f_i^{-1}(0)) \\ 0 & (\text{otherwise}) \end{cases}, f_{i\_1}(X) = \begin{cases} 1 & (X \in f_i^{-1}(1)) \\ 0 & (\text{otherwise}) \end{cases}, f_{i\_d}(X) = \begin{cases} 1 & (X \in f_i^{-1}(d)) \\ 0 & (\text{otherwise}) \end{cases} \tag{10}$$

Claim 9 (original): The Method to Reduce the Width of Graph according to Claim 8 wherein:

said system comprising "Means to Store Compatible Graphs" storing the compatible graph as a table of function node data, that is a table of column function labels of said each function node and the data of compatible edges connected to the function node, where the compatible graph is a graph that has nodes of column functions (function nodes), and wherein a pair of function nodes corresponding to column functions compatible each other with an edge or edges (compatible edges), and comprising:

said "Step to Generate Column Functions" in which, generates column functions corresponding to each edge of nodes at said height of the partition *lev* set in said "Step to Find Dividing Lines" from said node table stored in "Means to Store Node Table", and then generates said function node data labeled by column function labels corresponding to these column functions, and then stores in said "Means to Store Compatible Graphs";

a "Step to Generate Compatible Edges" which selects the pair of compatible column functions from the set of column functions corresponding to said each function node data, stored in said "Means to Store Compatible Graphs", and then adds compatible edge which connects function node data corresponding to these compatible column functions and these function node, and finally updates the function node data stored in said "Means to Store Compatible Graphs";

a "Step to Generate Cliques" covering all nodes in said compatible graph with the minimum number of complete subgraphs (cliques) and then generating clique data of the function node set contained in the clique; and

said "Step to Reconstruct Assigned BDD" which reconstructs said BDD_for_CF

by making some column functions to the identically assigned column functions by assigning constants to the *don't care(s)* of the column functions corresponding to each function node contained in the clique data produced by said "Means to Generate Cliques", and updates said node table in said "Means to Store Node Table".

Claim 10 (currently amended): The Method to Reduce the Width of Graph according to Claim 8 [[or 9]] wherein:

said "Step to Find Dividing Lines" to said "Step to Reconstruct Assigned BDD" is performed while changing said height *lev* of the partition sequentially from the height of the child node of the root node in BDD_for_CF represented by said node table stored in said "Means to Store Node Table", towards the lower height.

Claim 11 (currently amended): A Method for Logic Synthesis which generates look-up tables (LUTs) of the data for constructing logic circuits corresponding to said multiple-output logic function $F(X)$ from the BDD_for_CF of the multiple-output logic function $F(X)=(f_0(X), \cdots, f_{m-1}(X))$ with input variables $X=(x_1, \ldots, x_n)$ $(n \in N)$, in the system comprising:

"Means to Store Node Table" storing BDD_for_CF representing the characteristic function $\chi(X, Y)$ (where $Y=(y_0, \cdots, y_{m-1})$ $(m \geqq 2,\ m \in N)$ denotes the output variables of $F(X)$) defined in Equation (11), as the node table which is the table of the node data that consists of the labels of variables and pairs of edges $e_0(v_i)$ and $e_1(v_i)$, where said multiple-output logic function $F(X)=(f_0(X), \cdots, f_{m-1}(X))$ is a completely specified function, the labels of variables are labels given to the variables z_i $(z_i \in (X \cup Y))$ corresponding to said

non-terminal node v_i in the BDD_for_CF, and a pair of edges e_0 (v_i) and e_1 (v_i) that points the next transition child node(s) when the input values of z_i ($z_i \in (X \cup Y)$) are 0 and 1; and

"Means to Store LUTs" storing said LUTs, and comprising:

a "Step to Find Dividing Lines" setting the height of the partition lev which partitions BDD_for_CF represented by said node table stored in said "Means to Store Node Table";

a "Step to Reduce by Shorting" executing shorten-processing that is the processing to replace the edge $e_c(v_k)$ that points the node v_j among two edges $e_0(v_k)$ and $e_1(v_k)$ of the parent node v_k of the node v_j, by the edge $e_b(v_j)$ other than the edge $e_a(v_j)$ of the node v_j, in the case that the terminal node related to $\chi(X,Y)=0$ pointed by the edge $e_a(v_j)$ of either edge $e_0(v_j)$ or $e_1(v_j)$ of the node v_j, about the node data of the node v_j related to the variable $y_r(\in Y)$ representing the output and the parent node v_k of the node v_j, where the nodes v_j and v_k are contained in the subgraph B_0 including the root node among the node data of non-terminal nodes stored in said "Means to Store Node Table", in the case of partitioning BDD_for_CF to the two subgraphs B_0 and B_1 at the partition line in said height of the partition lev;

a "Step to Measure the Width of BDDs" which counts the number of the edges that point the child nodes of the non-terminal nodes, whose height is smaller than said height of the partition lev, among the edges which are the non-terminal nodes in BDD_for_CF to which said shorten-processing by said "Means to Reduce by Shorting" is applied, and which belong to the non-terminal nodes whose height is larger than said height of the partition lev (where the edges pointing the same node is counted as one, and the edges pointing the constant 0 are ignored), and produces the number at the partition

line in said height of the partition *lev* as the width *W*;

a "Step to Count the Intermediate Variables" counting the number of the intermediate variables *u* by Equation (12), using the width *W*;

a "Step to form LUT" which generates LUTs from the node data and stores said LUTs in said "Means to Store LUTs", for the non-terminal nodes which belong to the subgraph B_0 including the root node, among the non-terminal nodes stored in said "Means to Store Node Table", in the case of partitioning said height

a "Step to Reconstruct BDD" which generates a binary tree which has the same number of control inputs as the number of intermediate variables *u* which is calculated by said "Means to Compute the Intermediate Variables", and reconstructs the BDD_for_CF by replacing the node data of non-terminal nodes in subgraph B_0 of BDD_for_CF stored in said "Means to Store Node Table" with the node data representing said binary tree, and updates the node table stored in said "Means to Store Node Table" by the node data of the non-terminal nodes in said reconstructed BDD_for_CF[[.]] ,

[Equation 11]

$$\chi(X,Y) = \bigwedge_{i=0}^{m-1} (y_i \equiv f_i(X)) \qquad (11)$$

[Equation 12]

$$u = \lceil \log_2 W \rceil \qquad (12)$$

Claim 12 (currently amended): The Method for Logic Synthesis according to Claim 11

wherein said "Means to Store Node Table" stores the BDD_for_CF as a node table, where said BDD_for_CF is a graph that represents the characteristic function $\chi(X,Y)$ (where $Y=(y_0, \cdots, y_{m-1})$ ($m \geqq 2, m \in N$) denotes the output variables of $F(X)$) defined in Equation (13), with said multiple-output logic function $F(X)=(f_0(X), \cdots, f_{m-1}(X))$ of an incompletely specified function including *don't care* in outputs, said node table is the table of the node data that consists of the labels of variables and pairs of edges $e_0(v_i)$ and $e_1(v_i)$, said labels of variables are labels given to the variables z_i ($z_i \in (X \cup Y)$) corresponding to said non-terminal node v_i in the BDD_for_CF, and said pair of edges $e_0(v_i)$ and $e_1(v_i)$ points the next transition child node(s) when the values of z_i ($z_i \in (X \cup Y)$) are 0 and 1[[.]],

[Equation 13]

$$\chi(X,Y) = \bigwedge_{i=0}^{m-1} \{\bar{y}_i f_{i\_0} \vee y_i f_{i\_1} \vee f_{i\_d}\} \tag{13}$$

where $f_{i\_0}, f_{i\_1}, f_{i\_d}$ are the OFF-function, the ON-function and the DC-function defined in Equation (14), respectively[[.]],

[Equation 14]

$$f_{i\_0}(X) = \begin{cases} 1 & (X \in f_i^{-1}(0)) \\ 0 & (\text{otherwise}) \end{cases}, f_{i\_1}(X) = \begin{cases} 1 & (X \in f_i^{-1}(1)) \\ 0 & (\text{otherwise}) \end{cases}, f_{i\_d}(X) = \begin{cases} 1 & (X \in f_i^{-1}(d)) \\ 0 & (\text{otherwise}) \end{cases} \tag{14}$$

Claim 13 (original): The Method for Logic Synthesis according to Claim 12 which reduces the width of BDD_for_CF represented by said node table stored in said "Means to Store Node Table", by the Method to Reduce the Width of Graph according to any one of Claims 8 to 10, and then updates said node table stored in said "Means to Store Node Table ", and then performs from said "Step to Find Dividing Lines" to said "Step to Reconstruct BDD".

Claim 14 (currently amended): The Method for Logic Synthesis according to ~~any one of Claims 11 to 14~~ claim 11 wherein; performing the following three steps after performing from said "Step to Find Dividing Lines" to said "Step to Reconstruct BDD";

a "Step to Decide Ordering of Output Variables" deciding the order π of elements of said multiple-output logic function $F(X)$ to minimize the value of T represented in Equation (15), where $\pi=(\pi[0], \cdots, \pi[m-1])$ ($\pi[i]=j$ represents that f_j is the i'th element) is the order of the logic functions $f_0(X), \cdots, f_{m-1}(X)$ that are elements of said multiple-output logic function $F(X)$, and $supp(f_j)$ is the set of the input variables that influence the logic function f_j ($\in F(X)$);

a "Step to Decide Ordering of all the Variables" deciding the order of the variables $y_j(\in Y)$ representing the outputs and input variables x_i ($\in X$) in the order P that satisfies Equation (16); and

a "Step to Generate BDDs" which generates node data of the BDD_for_CF according to the order P decided in said "Means to Decide the Ordering of all the Variables", and then stores in said "Means to Generate BDDs"[[.]] ,

[Equation 15]

$$T = \sum_{k=0}^{m-1} \left| \bigcup_{l=0}^{k} \mathrm{supp}(f_{\pi[l]}) \right| \tag{15}$$

[Equation 16]

$$P = \left(\mathrm{supp}(f_{\pi[0]}), y_{\pi[0]}, \mathrm{supp}(f_{\pi[1]}) - \mathrm{supp}(f_{\pi[0]}), y_{\pi[1]}, \mathrm{supp}(f_{\pi[2]}) - \left(\sum_{k=0}^{1} \mathrm{supp}(f_{\pi[k]}) \right), y_{\pi[2]}, \right.$$
$$\left. \cdots, \mathrm{supp}(f_{\pi[m-1]}) - \left(\sum_{k=0}^{m-2} \mathrm{supp}(f_{\pi[k]}) \right), y_{\pi[m-1]} \right) \tag{16}$$

Claim 15 (currently amended): A computer program that implements the Method to Reduce the Width of Graph according to ~~any one of Claims 8 to 10~~ claim 8.

Claim 16 (currently amended): A computer program that implements the Method for Logic Synthesis according to ~~any one of Claims 11 to 15~~ claim 11.

Claim 17 (canceled):